REMARKS

The rejections presented in the Office Action dated September 13, 2004 have been considered. New claims 13-20 are added to claim the invention in alternative language, and claims 1-20 remain pending in the application. Reconsideration and allowance of the application are respectfully requested.

Claims 1-3, 6, 7 and 12 stand rejected under 35 USC §102(b) as being anticipated by US patent 5,822,787 to Zucker. (hereinafter "Zucker"). The rejection is respectfully traversed because the Office Action fails to show that all the limitations of the claims are taught by Zucker.

Claim 1 includes limitations of and related to identifying linkage code segments in the executable program, wherein each linkage code segment is associated with a call to a procedure in the shared library, reads procedure address information from a linkage table, and transfers control to an associated one of the procedures; reading the address information from the linkage table; generating respective substitute code segments for the linkage code segments, each substitute code segment having references to the linkage table replaced by direct loads of the address information without reference to the linkage table; and executing the substitute code segments instead of the linkage code segments. Various ones of these limitations are not shown to be taught by Zucker.

For example, nowhere is it shown that Zucker "identifies" linkage code segments. The cited section simply shows that Zucker's program has a procedure linkage table. There is no apparent need in Zucker for the claimed "identifying" of a linkage code segment.

Zucker is not shown to teach the claimed substitute code segments. Specifically, a substitute code segment has references to the linkage table made by the linkage code segment replaced by direct loads of the address information. It is respectfully submitted that none of Zucker's teachings as cited in the Office Action have any replacement of references to the linkage table. In fact, in Zucker's approach, the procedure linkage table continues to be referenced after having been updated as further explained below.

Claim 1 further indicates that the substitute code segment has references to the linkage table replaced by direct loads. Thus, in the present invention the program will no longer reference the linkage table to transfer control to a procedure during program execution once the substitute code segment has been generated because the substitute code segment has a direct load of the address information for that procedure. The Office Action does not show that Zucker's approach eliminates references to the linkage table as claimed. In fact, Zucker appears to teach how the entries in a procedure linkage table are established and modified by a dynamic linker. Zucker's col. 12, l. 35 – col. 16 l. 58). These teachings clearly show that Zucker's dynamic linker is called to initialize entries in the procedure lookup table. After an entry with an address is established for a function call by the dynamic linker, subsequent calls to that function do not require again calling on the dynamic linker. Instead, the entry in the procedure lookup table is used to call the function (col. 15, l. 60-61). Thus, Zucker continues to use the procedure lookup table.

Claim 2 includes further limitations of allocating relocation address space for the executable program; and storing the substitute code segments in the relocation address space. These limitations are not shown to be taught by Zucker because Zucker is not shown to teach the claimed substitute code segment and usage thereof as explained for claim 1.

Claim 3 includes further limitations of, for each calling procedure that calls a called procedure in the shared library, identifying registers that are not used by the calling procedure; and generating in a substitute code segment instructions that store in the registers the address information read from the linkage table. The Office Action does not show that Zucker teaches these limitations. It is respectfully submitted that the cited col. 10, ll. 64-67, makes no apparent reference to identifying registers that are not used and the presently claimed use thereof. If the rejection is maintained, an explanation of how the limitations are being interpreted to cover the asserted teachings of Zucker is requested because Zucker is not understood to teach any identification of unused registers.

Claims 6 and 7 include limitations that refine the limitations described above for claims 1, and 3. Therefore, the Office Action does not show that claim 6 is anticipated by Zucker for at least the reasons set forth above.

Claim 12 is an apparatus claim having functional limitations comparable to those set forth above for claim 1. Therefore, Zucker is not shown to anticipate claim 12 for at least the reasons set forth above.

The Office Action does not establish that claims 1-3, 6, 7 and 12 are anticipated by Zucker. Further clarification is requested if the rejection is maintained. Otherwise the rejection should be withdrawn.

The Office Action fails to show that claims 4 and 5 are unpatentable under 35 U.S.C. § 103(a) over Zucker in view of "McInerney" (US patent 5,956,479 to McInerney et al.). The rejection is respectfully traversed because the Office Action fails to show that all the limitations are suggested by the references, fails to provide a proper motivation for modifying the teachings of Zucker with teachings of McInerney, and fails to show that the combination could be made with a reasonable likelihood of success.

The Office Action cites McInerney's FIGs. 3 and 4 as suggesting the limitations of claim 4, including annotating linkage code segments in the executable program during compilation; placing breakpoints at locations of the linkage code segments as indicated by the annotations; and upon encountering a breakpoint at a linkage code segment, generating a substitute linkage code segment. However, McInerney describes FIG. 3 as an example of a Data Map used to map source program variables to information describing characteristics of the variables. FIG. 4 illustrates the entries and structure of a Statement Map (col. 7, ll .48-53). There is no apparent suggestion that a substitute code segment is generated upon encountering a breakpoint as claimed. Nor is there any apparent suggestion of applying annotations to linkage code segments. An explanation of specific ones of McInerney's elements are thought to teach these limitations is requested if the rejection is maintained.

The Office Action fails to provide a proper motivation for combining McInerney with Zucker. The alleged motivation for making the combination is that annotating and placing breakpoints at locations of linkage code in Zucker's system "would allow the user to perform debugging methods on the code, thereby increasing the quality of the overall code and system." This alleged motivation is inapplicable to the claimed invention.

First, it is unclear why one would include breakpoints for debugging Zucker when the claim is clearly directed to using the breakpoints to trigger generating the claimed substitute code segment. Zucker already has a mechanism to modify the procedure linkage table, and no evidence is provided to indicate that an alternative approach would be beneficial to Zucker.

Second, McInerney's system is for the development of complex computer programs (Abstract), whereas Zucker describes an application binary interface for interfacing a binary application program to a digital computer (Abstract). Zucker apparently describes an operationally correct and complete system. Thus, there is no apparent need for debugging of Zucker's system. Therefore, the alleged motivation is improper.

The rejection of claims 4-5 over the Zucker-McInerney combination should be withdrawn because the Office Action fails to show all the limitations are suggested by the combination, fails to provide a proper motivation for combining the references, and fails to show that the combination could be made with a reasonable likelihood of success.

The Office Action fails to show that claims 8-11 are unpatentable under 35 U.S.C. § 103(a) over Zucker in view of "Shaughnessy" (US patent 6,026,235 to Shaughnessy). The rejection is respectfully traversed because the Office Action fails to show that all the limitations are suggested by the references, fails to provide a proper motivation for modifying the teachings of Zucker with teachings of Shaughnessy, and fails to show that the combination could be made with a reasonable likelihood of success.

Shaughnessy is cited as suggesting the use of breakpoints at function entry points. However, the Office Action ignores the additional limitations of, upon encountering a breakpoint at the entry point of a procedure during program execution, searching for linkage code segments in the procedure, generating substitute linkage code segments for those identified in the procedure, and continuing with execution of the procedure. It is respectfully submitted that no teachings of either Zucker or Shaughnessy are cited as suggesting these additional limitations. Thus, the Office Action fails to show that all the limitations are suggested by the Zucker-Shaughnessy combination.

In addition, the alleged motivation for combining Shaughnessy with Zucker is unsupported by evidence, and is therefore, improper. The alleged motivation is that "it would have been obvious ... to place breakpoints at procedure entry points in the system disclosed by Zucker, which would allow the user to perform debugging methods on the code, thereby increasing the quality of the overall code and system." This alleged motivation simply repeats the alleged motivation stated for making the Zucker-McInerney combination. Therefore, this alleged motivation is improper for the reasons set forth above in the traversal of the Zucker-McInerney combination.

Claims 9-11 are allowable over the Zucker-McInerney combination for at least the reasons set forth above for claim 8, as well as the reasons set forth for claims 2 and 7.

The rejection of claims 8-11 over the Zucker-Shaughnessy combination should be withdrawn because the Office Action fails to show all the limitations are suggested by the combination, fails to provide a proper motivation for combining the references, and fails to show that the combination could be made with a reasonable likelihood of success.

Withdrawal of the rejection and reconsideration of the claims are respectfully requested. If the examiner has any questions or concerns, a telephone call to the undersigned is welcome. No extension of time is believed to be necessary for consideration of this response. However, if an extension of time is required, please consider this a petition for a sufficient number of months for consideration of this response.

Respectfully submitted,

CRAWFORD MAUNU PLLC
1270 Northland Drive, Suite 390
Saint Paul, MN 55120
(651) 686-6633

By: ______________________
Name: LeRoy D. Maunu
Reg. No.: 35,274